

In the Claims:

Please amend Claims 34, 36-37, 50, 53, 55-56, 62, and 72, cancel Claims 35, 38, 54, and 57, and add new Claims 73-74, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

1-33. (Canceled).

34. (Currently Amended) A method for rendering a graphical user interface (GUI), comprising:

providing for the representation of the GUI as a plurality of objects, wherein each object of the plurality of objects represents a control in the GUI, wherein each said object is associated with one or more states in a lifecycle of the control, wherein the state information of each said object can be persisted using a state management persistence mechanism;

receiving, from a client, a request to render the GUI, wherein the request contains the state information for each said object;

generating a logical hierarchy for the plurality of objects, in a container, using metadata, wherein the metadata is based on one or more definitions in a page description language, and wherein the metadata includes information about properties, events, and model binding that have values set in page descriptions; [[and]]

using a control state reader chain to read the state information in the request, wherein the control state reader chain contains one or more control state readers, wherein each control state reader operates to read the state information for at least one said object, and wherein the container can find a corresponding control state reader for a particular control;

updating the logical hierarchy based on the state information; and
generating a response based on the updated logical hierarchy and returning the response to the client

~~using an interchangeable lifecycle driver to drive the logical hierarchy through a sequence of states, wherein the interchangeable lifecycle driver isolates lifecycle driver implementation details from a container of the logical hierarchy and allows different lifecycle implementations to be interchanged.~~

35. (Canceled).

36. (Currently Amended) The method of claim [[35]] 34 wherein:
the request in a hypertext transfer protocol (HTTP) request.
37. (Currently Amended) The method of claim [[35]] 34 wherein:
the request originates from a [[Web]] web browser.
38. (Canceled).
39. (Previously Presented) The method of claim 34, further comprising:
allowing a first object in the logical hierarchy to respond to an event raised by a second object in the logical hierarchy.
40. (Previously Presented) The method of claim 34, further comprising:
allowing an object in the logical hierarchy to use an interchangeable persistence mechanism.
41. (Previously Presented) The method of claim 34, further comprising:
allowing an object in the logical hierarchy to use an interchangeable rendering mechanism.
42. (Previously Presented) The method of claim 34 wherein:
in the providing step, an object can represent one of: button, text field, menu, table, window, window control, title bar, pop-up window, check-box button, radio button, window frame, desktop, shell, head, body, header, footer, book, page, layout, placeholder, portlet and toggle button.
43. (Previously Presented) The method of claim 34, further comprising:
allowing an object in the logical hierarchy to inherit a theme from a parent object.
44. (Previously Presented) The method of claim 34 further comprising:

providing a theme that specifies the appearance and/or functioning of an object of the logical hierarchy in the GUI.

45. (Previously Presented) The method of claim 34 further comprising:

rendering a first object in the logical hierarchy in parallel with a second object in the logical hierarchy.

46. (Previously Presented) The method of claim 34 further comprising:

specifying a theme for the logical hierarchy in whole or in part by a properties file.

47. (Previously Presented) The method of claim 46 wherein:

in the specifying step, the properties file can include at least one of: 1) cascading style sheet; 2) Java Server Page; 3) Extensible Markup Language; 4) text; 5) Hypertext Markup Language; 6) Extensible Hypertext Markup Language; 7) JavaScript; and 8) Flash MX.

48. (Previously Presented) The method of claim 46 wherein:

in the specifying step, the properties file can specify at least one image.

49. (Previously Presented) The method of claim 34 wherein:

in the providing step, the GUI is part of a portal on the World Wide Web.

50. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

provide for the representation of the GUI as a ~~plurality~~ plurality of objects, wherein each object of the plurality of objects represents a control in the GUI, wherein each said object is associated with one or more states in a lifecycle of the control, wherein the state information of each said object can be persisted using a state management persistence mechanism;

receive, from a client, a request to render the GUI, wherein the request contains the state information for each said object;

generate a logical hierarchy for the plurality of objects in a container, using metadata, wherein the ~~meta-data~~ metadata is based on one or more definitions in a page description language, and wherein the ~~meta-data~~ metadata includes information about properties, events, and model binding that have values set in page descriptions; ~~[[and]]~~

use a control state reader chain to read the state information in the request, wherein the control state reader chain contains one or more control state readers, wherein each control state reader operates to read the state information for at least one said object, and wherein the container can find a corresponding control state reader for a particular control;

update the logical hierarchy based on the state information; and

generate a response based on the updated logical hierarchy and return the response to the client

~~use an interchangeable lifecycle driver to drive the logical hierarchy through a sequence of states, wherein the interchangeable lifecycle driver isolates lifecycle driver implementation details from a container of the logical hierarchy and allows different lifecycle implementations to be interchanged.~~

51. (Previously Presented) The machine readable medium of claim 50 further comprising instructions to:

allow one of the plurality of objects to respond to an event raised by another of the set of objects.

52. (Previously Presented) The machine readable medium of claim 50 further comprising instructions to:

allow a control to use an interchangeable persistence mechanism.

53. (Currently Amended) The machine readable medium of claim 50 further comprising instructions to:

allow a control to use have an interchangeable rendering mechanism.

54. (Canceled).

55. (Currently Amended) The machine readable medium of claim [[54]] 50 wherein:
the request in a hypertext transfer protocol (HTTP) request.

56. (Currently Amended) The machine readable medium of claim [[54]] 50 wherein:
the request originates from a [[Web]] web browser.

57. (Canceled).

58. (Previously Presented) The machine readable medium of claim 50 wherein:

an object of the plurality of objects can represent one of: button, text field, menu, table, window, window control, title bar, pop-up window, check-box button, radio button, window frame, desktop, shell, head, body, header, footer, book, page, layout, placeholder, portlet and toggle button.

59. (Previously Presented) The machine readable medium of claim 50 further comprising instructions to:

associate a theme with an object when the object is rendered.

60. (Previously Presented) The machine readable medium of claim 50 further comprising instructions to:

allow an object of the plurality of objects to inherit a theme from a parent object.

61. (Previously Presented) The machine readable medium of claim 50 further comprising instructions to:

provide a theme that specifies the appearance and/or functioning of a first object in the GUI.

62. (Currently Amended) The machine readable medium of claim 50 further comprising instructions to:

render [[an]] a first object of the plurality of objects according to a theme in parallel with other objects.

63. (Previously Presented) The machine readable medium of claim 50 further comprising instructions to:

specify a theme for the plurality of objects in whole or in part by a properties file.

64. (Previously Presented) The machine readable medium of claim 63 wherein:

to specify a theme for the plurality of objects, the properties file can include at least one of: 1) cascading style sheet; 2) Java Server Page; 3) Extensible Markup Language; 4) text; 5)

Hypertext Markup Language; 6) Extensible Hypertext Markup Language; 7) JavaScript; and 8) Flash MX.

65. (Previously Presented) The machine readable medium of claim 63 wherein:
to specify a theme for the plurality of objects, the properties file can specify at least one image.

66. (Previously Presented) The machine readable medium of claim 50 wherein:
the GUI is part of a portal on the World Wide Web, to provide for the representation of the GUI as a plurality of objects.

67. (Canceled).

68. (Previously Presented) The method of claim 34 wherein:
one of the plurality of objects is a desktop object and the desktop object contains one or more personalized views.

69. (Previously Presented) The method of claim 34, further comprising:
generating a page implementation class to render the GUI in concert with the logical hierarchy.

70. (Previously Presented) The method of claim 34, further comprising:
mapping one or more tag extensions into the logical hierarchy when the logical hierarchy is rendered, wherein the one or more tag extensions represent at least one control in the logical hierarchy, and wherein at least one tag extension can locate a metadata description of the logical hierarchy and create the logical hierarchy.

71. (Previously Presented) The method of claim 34, further comprising:
implementing the logical hierarchy as a control tree.

72. (Currently Amended) The method of claim 71, further comprising:
using a streaming control tree factory to create the control tree from an XML stream,
wherein the XML stream is obtained from multiple resources, wherein the streaming control tree

factory can map each user into an individual control stream and regenerate the control tree if the XML stream changes.

73. (New) The method of claim 34, further comprising:

using an interchangeable lifecycle driver to drive the logical hierarchy through a sequence of states, wherein the interchangeable lifecycle driver isolates lifecycle driver implementation details from a container of the logical hierarchy and allows different lifecycle implementations to be interchanged.

74. (New) The method of claim 34, further comprising:

specifying the control state reader chain in an external text file that is parsed at runtime.